

AMENDMENTS TO THE CLAIMS

1. (currently amended) A computer implemented method for communicating data ~~in a clustered computing system~~, the method comprising:
~~detecting an occurrence of receiving~~ an initial indication that an event was generated at a first node of ~~[[the]] a~~ system;
~~detecting an occurrence of receiving~~ one or more subsequent indications that the event was generated events at the first node of the system;
determining that ~~the information about from~~ the initial event indication is identical to ~~the information about from~~ said one or more subsequent indications events;
in response to determining that the information ~~about from~~ the initial event indication is identical to the information ~~about from~~ said one or more subsequent ~~events indications~~, coalescing the information from the initial indication with the information from said one or more subsequent indications into a coalesced notification; and
~~appending, onto an existing message, a notification that includes information that describes a single instance of an event selected from a set of events that consists of (a) said initial event; and (b) said one or more subsequent events;~~
propagating the coalesced notification to a receiving node;
~~wherein the message is destined to be propagated to the receiving node,~~
~~wherein the receiving node is not a node sending the message.~~
2. (cancelled)
3. (cancelled)
4. (cancelled)

5. (currently amended) The method of Claim 1, wherein said ~~elustered computing~~ system comprises a database management system.
6. (currently amended) The computing environment of Claim 1, wherein said ~~elustered computing~~ system comprises a shared-disk database system.
7. (currently amended) The computing environment of Claim 1, wherein said ~~elustered computing~~ system comprises a shared-cache parallel database management system.
8. (currently amended) The computing environment of Claim 1, wherein said ~~elustered computing~~ system comprises a shared-nothing database management system.
9. (currently amended) The computing environment of Claim 1, wherein said ~~elustered computing~~ system comprises a distributed database management system.
- 10-21. (cancelled)
22. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 1.
- 23-25. (cancelled)
26. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or

- more processors, causes the one or more processors to perform the method recited in Claim 5.
27. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 6.
28. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 7.
29. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 8.
30. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 9.
- 31-42. (cancelled)
43. (currently amended) A computer implemented method for communicating data in a ~~clustered computing system~~, the method comprising:
~~detecting an occurrence of~~ receiving an indication that an event was generated at a first node of ~~[[the]] a system[[.]]~~;
receiving one or more subsequent indications that the event was generated at the first node of the system;

determining if the information about said event is identical to another previously occurring event;
determining that information from said indication is identical to information from said one or more subsequent indications;
in response to determining that the information from the initial indication is identical to the information from said one or more subsequent indications, coalescing the information from the initial indication with the information from said one or more subsequent indications into a coalesced notification;
appending onto an existing message [[a]] the coalesced notification that describes a single instance of said event, wherein the message was destined to be propagated to a receiving node that is not a node sending the message; and
propagating the coalesced notification to the receiving node.

44. (cancelled)
45. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 43.
46. (cancelled)
47. (currently amended) A computer implemented method comprising the computer-implemented steps of:
receiving an initial indication that an event was generated at a local node;
receiving, at a database server that is executing on a second node in a distributed system, a message that (a) was transmitted by a first node in the distributed system and (b) subsequently had appended thereon information describing [[an]] the event;

- retrieving the information describing the event from the local node where said event occurred; [[and]]
invoking concurrency control techniques to control concurrent access to a shared-memory event buffer from processes that propagate messages to subscriber nodes and processes that generate events.
48. (cancelled)
49. (currently amended) The method of Claim 47, wherein said method further comprises:
maintaining information that describes a plurality of events,
coalescing the information that describes a plurality of events, wherein the information that describes a plurality of events may be coalesced for ~~the same~~ an identical event into a single coalesced event notification.
50. (original) The method of Claim 47, wherein the method further comprises:
maintaining the information that describes a plurality of events.
51. (currently amended) The method of Claim 47, wherein the method further comprises step of:
maintaining information that describes the plurality of events in [[a]] the shared-memory event buffer.
52. (previously presented) The method of Claim 47, wherein the method further comprises the step of:
maintaining information that describes the plurality of events in a circular buffer.
53. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or

- more processors, causes the one or more processors to perform the method recited in Claim 47.
54. (cancelled)
55. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 49.
56. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 50.
57. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 51.
58. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 52.
59. (original) A computer apparatus comprising:
a processor; and
a memory coupled to the processor, the memory containing one or more sequences of instructions for event notification in a clustered computing environment, wherein execution of the one or more sequences of instructions by the processor causes the processor to perform the method of Claim 1.

60. (new) The method of Claim 1, wherein the step of propagating comprises:
appending, onto an existing message, the coalesced notification that includes
information that describes a single instance of said event;
wherein the message is destined to be propagated to the receiving node, wherein the
receiving node is not a node sending the message.
61. (new) The method of Claim 60, where the step includes piggybacking
the coalesced notification on a message that is otherwise unrelated to the
event.
62. (new) The method of Claim 60, wherein the method further comprises:
setting an identifier indicating that the information describing an identical event is to
be appended onto a message and propagated to a particular node.
63. (new) The method of Claim 60, wherein the message has a fixed size, and the
method further comprises:
appending additional information that describes additional events onto existing
message traffic until free space in the fixed-size message is filled.
64. (new) The method of Claim 60, wherein the method further comprises
placing the information describing an identical event in a queue.
65. (new) The method of Claim 64, wherein the queue includes at least a priority
queuing mechanism in order to determine a priority for events such that high priority
events would supercede a low priority event in the queue.
66. (new) The method of Claim 60, wherein an in-memory hash index is used to
determine if an event exists in a shared-memory event buffer.

67. (new) The method of Claim 66, wherein the shared-memory event buffer has a fixed size.
68. (new) The method of Claim 60, wherein the method further comprises:
partitioning a shared-memory event buffer;
generating an event buffer entry of the shared memory event buffer;
placing an event identifier into the event buffer entry; and
inserting the information describing an identical event into the event buffer entry.
69. (new) The method of Claim 68, the method further comprises if between a fastest head pointer and a tail pointer there does not exist a buffer entry in the shared memory event buffer for an identical event, generating a new event buffer entry, and the inserting further comprises inserting the information describing said identical event into the new event buffer entry.
70. (new) The method of Claim 68, wherein the inserting comprises:
if between a fastest head pointer and a tail pointer there exists a buffer entry in the shared memory event buffer for the identical event, updating the buffer entry so that the buffer entry represents the subsequent occurrence.
71. (new) The method of Claim 68, further comprising
using a round robin method and the shared memory event buffer to determine to which existing message to append the information describing an identical event.
72. (new) The method of Claim 60, wherein the method further comprises the step of:
maintaining information that describes a plurality of events.
73. (new) The method of Claim 72, wherein the method further comprises the step of :

maintaining information that describes the plurality of events in a shared-
memory event buffer.